CHAM: ACTION RECOGNITION USING CONVOLUTIONAL HIERARCHICAL ATTENTION MODEL

Shiyang Yan[a], Jeremy S. Smith[b], Wenjin Lu[a] and Bailing Zhang[a]

[a]Computer Science and Software Engineering Department,
Xi'an Jiaotong-Liverpool University
[b]University of Liverpool

ABSTRACT

Recently, the soft attention mechanism, which was originally proposed in language processing, has been applied in computer vision tasks like image captioning. This paper presents improvements to the soft attention model by combining a convolutional LSTM with a hierarchical system architecture to recognize action categories in videos. We call this model the Convolutional Hierarchial Attention Model (CHAM). The model applies a convolutional operation inside the LSTM cell and an attention map generation process to recognize actions. The hierarchical architecture of this model is able to explicitly reason on multi-granularities of action categories. The proposed architecture achieved improved results on three publicly available datasets: the UCF sports dataset, the Olympic sports dataset and the HMDB51 dataset.

***Index Terms*—** Action recognition, Soft attention, Convolutional LSTM, CNN, Hierarchical Architecture

1. INTRODUCTION

Action recognition in video has been a popular yet challenging task which has received significant attention by the computer vision society [1] [2]. The potential applications of action recognition include video retrieval (i.e., YouTube videos), intelligent surveillance and interactive systems. Compared with action recognition from still images, the temporal dynamics provides an important clue to recognize human actions in videos.

Among the proposed models to capture the spatial-temporal transition in videos, Recurrent Neural Networks (RNN) are the preferred candidate due to the special internal memory being able process arbitrary sequences of inputs. A RNN is a class of artificial neural network where connections between the units form a directed cycle, and the internal state created from the network allows it to exhibit dynamic temporal behavior. Much research was conducted on RNNs in the 80s [3] [4] for time-series modeling, however this, was hampered for a long period by the difficulties of training, particularly the vanishing gradient problem [5]. Roughly speaking, the error gradients would vanish exponentially quickly with the size of the time lag between important events, which makes training very difficult. To mitigate this problem, a class of models with a long-range learning capability, called Long Short-Term Memory (LSTM), was introduced by Hochreiter, et al [6]. LSTM consists of memory blocks, with each block containing self-connected memory units to learn when to forget previous hidden states and when to update hidden states given new information. It has been verified that complex temporal sequences can be learnt by the LSTM [7].

LSTM has a close relationship with attention models in vision research and natural language processing (NLP). Human perception is characterized by an important mechanism of focusing attention selectively on different parts of a scene which has long been an important subject in the vision community. An attention model can be built using LSTM on top of image features to decide when the model should focus on certain parts of the image sequentially. In NLP, the attention model was proposed for sequence to sequence training in machine translation [8], where two types of attention model have been studied, hard attention and soft attention. Soft attention is deterministic and can be trained using back-propagation [9]. Soft attention was then extended to the image captioning task [9] since image captioning can be essentially considered as image to language translation. Sharma, et al.[10] used pooled convolutional descriptors with soft attention based models for action recognition and achieved good results. Continuing the previous research, we investigated the soft attention model in the action recognition context, and propose several improvements. Normally the LSTM is built on fully connected layers in which all the state-to-state transitions are matrix multiplication. This structure does not take spatial information into account. Xingjian, et al.[11] proposed convolutional LSTM in which all the transitions are convolutional operations. Following [11], we improved the soft attention model based on convolutional LSTM.

In real world applications, an action is usually composed of a set of sub-actions. For instance, jump shooting basketball often consists of three sub-actions- jumping, shooting and landing. This is a typical hierarchical structure in terms of motion dynamics. In other words, actions are composed

of multiple granularities. A straightforward way to model the layered action would be a hierarchical structure. Following [12] in which a Hierarchical Attention Networks (HAN) was proposed, we applied HAN with a convolutional LSTM to recognize multiple granularities of layered action categories. The proposed model can be termed CHAM which means Convolutional Hierarchical Attention Model.

Our main contributions can be summarized as follows:

(1) As deep features from CNNs preserve the spatial information, we improved the soft attention model by introducing convolutional operations inside the LSTM cell and attention map generation process to capture the spatial layout.

(2) To explicitly capture layered motion dependencies of video streams, we built a hierarchical two layer LSTM model for action recognition.

(3) We tested our model on three widely applied datasets, the UCF sports dataset [13], the Olympic dataset [14] and the HMDB51 dataset [15] with improved results on other published work.

2. SOFT ATTENTION MODEL FOR VIDEO ACTION RECOGNITION

2.1. Convolutional Soft Attention Model

LSTM was proposed by Hochreiter, et al [6] in 1997 and have subsequently been refined. LSTM is able to avoid the gradient vanishing problem and implements long term memory by incorporating memory units that allow the network to learn when to forget previous hidden states and when to update hidden states. The input, forget and output gates are composed of a sigmoid activation layer and matrix multiplication to define how much information flow should be passed to the next time-step. All the parameters in the gates can be learnt in the training process.

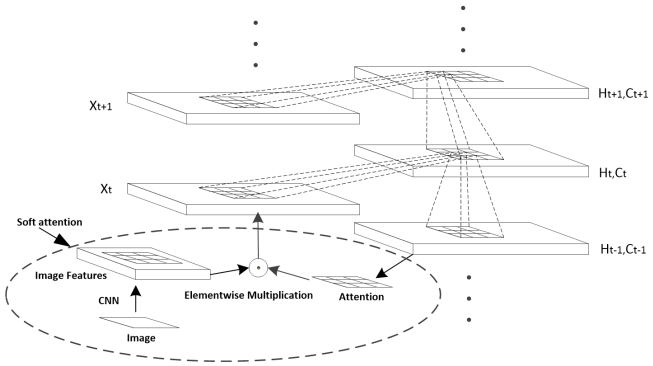


Fig. 1. The input-to-state, and state-to-state transition are all convolutional, the attention map is also generated by convolution. The soft attention mechanism is to elementwise multiply attention map with image features and forward to the convolutional LSTM at each time step.

Following the idea of [11], we replaced the state-to-state transitions in LSTM with convolutional operations which are illustrated in Fig.1. In Fig.1, the dashed lines indicate the convolution operations, all the input-to-state and state-to-state transitions are replaced with convolutions. Moreover, the attention map is derived from the hidden layer of the LSTM also using convolutional operations. The attention map will be elementwise multiplied with image features to select the most informative regions to focus on.

Our soft attention model is built upon deep CNN features. The features were extracted from the last convolutional layer from a CNN model trained on the ImageNet [16] database. The last convolutional features would have shape of $K \times K \times D$. We consider the features as K^2 number of D feature vectors in which each of the feature vectors represent overlapping receptive fields in the input image and our soft attention model choose to focus on different regions in each time step.

Letting $\sigma(x) = (1 + e^{-x})^{-1}$ be the sigmoid non-linear activation function and $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ be the tangent non-linear activation function, the convolutional LSTM model with soft attention follows these updating rules:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \quad (3)$$

$$g_t = \sigma(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \quad (5)$$

$$h_t = o_t \cdot \phi(c_t) \quad (6)$$

Here, i_t, f_t, o_t are the input, forget and output gates of the LSTM model, respectively. They are calculated according to Equations1 - 3. c_t is the cell memory while h_t is the hidden state of the LSTM model. A $*$ indicated the convolution operation. $W_\sim, b_\sim$ are convolutional weights and bias, respectively. The multiplication operations are all elementwise multiplication. x_t is the input to the LSTM model at each time step. It can capture the attention information given image features and the hidden state of LSTM from the last time step. Assuming F_t is the frame level image features which are $K \times K \times D$ dimension, x_t, the attention map on image features can be computed as follows:

$$x_t = l_t^{ij} \cdot F_t \quad (7)$$

$$l_t^{ij} = SOFTMAX(W_z * \phi(W_{ha} * h_{t-1} + W_{xa} * x_t + b_a)) \quad (8)$$

l_t^{ij} indicates the attention value of each region which is dependent on the hidden state of the last time step and the input image features of this time step. i, j means the horizontal and vertical position of the attention map, respectively. We achieve this by simple weighting of the image features with attention values to preserve the spatial information instead of

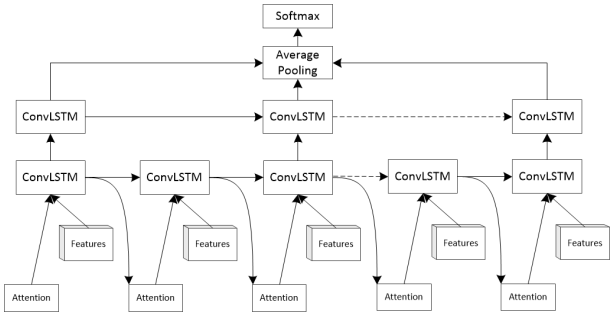


Fig. 2. This is a two layered hierarchical model in which the first attention layer reasons on each frame and the second layer skips several steps. The outputs from the two layers are concatenated and forwarded to the average pooling layer before the softmax classifier.

getting the expectation of image features as in [9]. This is essentially a type of amplification of the 'attention' location of features for the classification at hand. In practice, the hidden state of the last time step and input features are convolved by maps W_{ha} and W_{xa} respectively before passing to a softmax activation layer as in Equation 8. The softmax values can be considered as the importance of each region in the image features for the model to pay attention.

Finally, the model applied the cross-entropy loss for action classification.

$$LOSS = -\sum_{t=1}^{T}\sum_{i=1}^{C} y_{t,i} log(\hat{y}_{t,i}) \tag{9}$$

where y_t is the label vector, $\hat{y}_t$ is the classification probabilities at time step t. T is the number of time steps and C is the number of action categories.

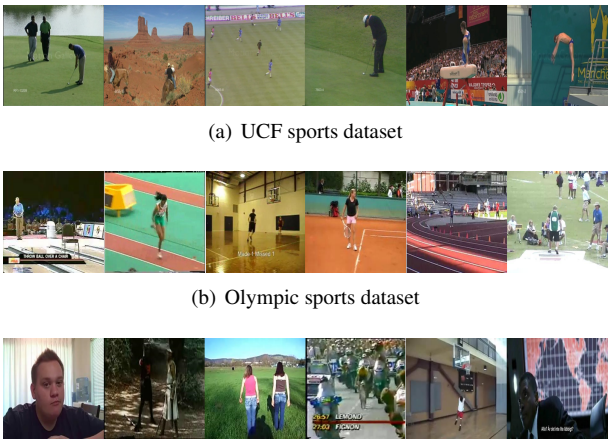
2.2. Hierarchical Architecture

As previously introduced, the hierarchical architecture of our CHAM is to capture layered motion dependencies. Fig. 2 illustrates the system structure of our hierarchical model. The first layer is the attention layer and is also able to reason on the more fine-grained properties of the temporal dependency. The second layer directly connects with first layer but skip several steps in order to catch the coarse granularity of the motion information. Then the output features of the first layer and second layer are concatenated before forwarding to the fully connected layers and an average pooling layer. Then a softmax classifier is connected to generate the results.

3. EXPERIMENTS

3.1. Datasets Introduction

The approach was evaluated on three datasets, namely the UCF sports [13], the Olympic sports [14] and the more difficult HMDB51 [15]. Fig.3 provides some examples of the three datasets. The UCF sports dataset contains actions collected from various sports on broadcast channels such as ESPN and the BBC. This dataset consists of 150 videos and with 10 different action categories present. The Olympic sports dataset was collected from YouTube sequences [14] and contains 16 different sports categories with 50 sequences per class. The full name of HMDB51 is Human Motion Database and it provides three train-test splits each consisting of 5100 videos. These clips are labeled with 51 action categories. The training set for each split has 3570 videos and the test set has 1530 videos.



(a) UCF sports dataset

(b) Olympic sports dataset

(c) HMDB51 dataset

Fig. 3. Some examples from the datasets used in this paper.

For the UCF sports dataset, we manually divide the dataset into a training and a testing set. We used 75% for training, and 25% for testing. We then report the frame-level accuracy based on the testing dataset.

For the Olympic sports dataset, we used the original training-testing split with 649 clips for training and 134 clips for testing. Following [14], we evaluated the Average Precision (AP) of each category on this dataset.

When evaluating our methods on HMDB51, we follow the original training-testing split and test the accuracy of each split. As [10] has the results of the conventional soft attention scheme, we only test the performance of our methodologies.

3.2. Implementation Details

Firstly, we extracted frame-level CNN features using MatConvNet [17] based on Residual-152 Networks[18] trained on the ImageNet [16] dataset. The images were resized to 224×224, hence the dimension of each frame-level features is 7×7×2048.

Then CHAM was built using the Theano [19] platform. We use a convolutional kernel size of 3×3 for state-to-state transition in LSTM and a 1×1 convolutional kernel for attention map generation to capture spatial information of the

Table 1. Accuracy on UCF sports

Methods	Accuracy
FC-Attention [10]	70%
Conv-Attention(Ours)	72%
CHAM(Ours)	**74%**

Table 2. AP on Olympics sports

Class	Vault	Triple Jump	Tennis serve	Spring board	Snatch
FC-Attention [10]	97.0%	88.4%	52.3%	60.0%	23.2%
Conv-Attention(Ours)	97.0%	94.0%	49.8%	66.4%	26.1 %
Conv-HAN(Ours)	97.0%	98.9%	49.5%	69.2%	47.8%
Shot put	Pole vault	Platform 10m	Long jump	Javelin Throw	High jump
67.4%	69.8%	84.1%	100.0%	89.6%	84.4%
60.0%	100.0 %	86.0%	98.0%	87.9%	80.0%
79.8%	60.8%	89.7%	100%	95.0%	78.7%
Hammer throw	Discus throw	Clean and jerk	Bowling	Basketball layup	mAP
38.0%	100.0%	76.0%	60.0%	89.8%	73.7%
36.6%	97.8%	100.0%	46.8%	81.2%	75.5%
37.9%	97.0%	84.8%	46.7%	89.1%	**76.4%**

CNN features. When the kernel size is 3×3, to ensure the states of LSTM in different time step have the same number of columns and rows as inputs, padding is needed before the convolution operation starts. All these convolutional kernels have 512 channels. A dropout is also applied on the output before being fed to the final softmax classifier with a ratio of 0.5.

Also, to carry out comparative studies, a convolutional attention model (Conv-Attention) using only one layer of the convolutional LSTM was built. The fully connected attention model (FC-Attention) based soft attention [10] was also implemented as a baseline approach. We set the matrix dimension of state-to-state transition in the fully connected LSTM as 512. The soft attention mechanism followed the setting in [10]. All the experiments were conducted using and an NVIDIA TITAN X.

For the network training, we applied mini-batch size of 64 samples at each iteration. For each video clip, the FC-Attention and Conv-Attention networks randomly selected 30 frames for training while CHAM seleted 60 frames for training with a second LSTM layer skip every 2 time steps. We applied the back propagation algorithm through time and Adam optimizer [20] with a learning rate of 0.0001 to train the networks. The learning rate was changed to 0.00001 after 10,000 iterations.

3.3. Results and Discussion

The results on the UCF sports dataset can be seen in Table.1. The Conv-Attention which apply convolutional LSTM for soft attention achieves 72% accuracy on the UCF sports dataset while FC-attention has 70% accuracy. CHAM has the highest accuracy of 74% which indicates that the hierarchical architecture is able to further improve on the system performance.

We then recorded the AP value of our methods on the Olympics sports dataset as shown in Table.2. The Conv-Attention method has a mean AP value of 75.5% which is higher than the FC-attention performance (73.7%). Similarly, the improvement brought by the hierarchical architecture is also validated on this dataset, with a 76.4% mean AP value achieved by the proposed CHAM model. The hierarchical model are especially good at long-term action categories, for instance, 'Snatch' and 'Javelin Throw' on which the CHAM method leads the other approaches by a large margin.

Table 3. Accuracy on HMDB51

Methods	Accuracy
FC-Attention [10]	41.3%
Conv-Attention(Ours)	42.2%
CHAM(Ours)	**43.4%**

Table 4. Comparison with related methods on HMDB51

Methods	Accuracy	Spatial Image Only	Fine-tuning
Softmax Rgression [10]	33.5%	Yes	No
Spatial Convolutional Net [2]	40.5%	Yes	Yes
Trajectory-based modeling [21]	40.7%	No	No
Average pooled LSTM [10]	40.5%	Yes	No
FC-Attention [10]	41.3%	Yes	No
ConvALSTM [22]	43.3%	Yes	Yes
CHAM(Ours)	**43.4%**	Yes	No

The results on the HMDB51 dataset can be seen in Table.3. Similar observations can be made: the Conv-Attention has a higher accuracy value of 42.2% and the hierarchical architecture(CHAM) added another 1.2% gain to the final result, which is 43.4%.

Table.4 shows the comparison results on the HMDB51 dataset. From the table, the following observation can be made:

(1) Our CHAM method outperformed most of the previous methods which are only based on spatial image features.

(2) Even though our CNN model was not fine-tuned, the results still remain competitive compared with many approaches which had applied fine-tuning.

(3) The proposed model shows good potential to achieve better results. Future work can be undertaken by fine-tuning the CNN model on a specific dataset.

4. CONCLUSION

In this paper we proposed a novel model: CHAM. This is achieved by applying convolutional LSTM, a novel RNN model, for the implementation of soft attention mechanism and a hierarchial system architecture for action recognition. The convolutional LSTM is able to catch the spatial layout of the CNN features while the hierarchical system architecture can fuse information on the temporal dependencies from multiple granularities of the dataset. At last, the CHAM method was tested on three widely used datasets, the UCF sports dataset, the Olympic sports dataset and the HMDB51 dataset, with improved results.

5. REFERENCES

[1] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[2] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.

[3] Jeffrey L Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[4] Paul J Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural Networks*, vol. 1, no. 4, pp. 339–356, 1988.

[5] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[6] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *arXiv preprint arXiv:1502.03044*, vol. 2, no. 3, pp. 5, 2015.

[10] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[11] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[12] Yilin Wang, Suhang Wang, Jiliang Tang, Neil O'Hare, Yi Chang, and Baoxin Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[13] Mikel Rodriguez, "Spatio-temporal maximum average correlation height templates in action recognition and video summarization," 2010.

[14] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European conference on computer vision*. Springer, 2010, pp. 392–405.

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[17] Andrea Vedaldi and Karel Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 689–692.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[19] James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, David Warde-Farley, Ian Goodfellow, Arnaud Bergeron, et al., "Theano: Deep learning on gpus with python," in *NIPS 2011, BigLearning Workshop, Granada, Spain*. Citeseer, 2011.

[20] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu, and Chong-Wah Ngo, "Trajectory-based modeling of human actions with motion reference points," in *European Conference on Computer Vision*. Springer, 2012, pp. 425–438.

[22] Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees GM Snoek, "Videolstm convolves, attends and flows for action recognition," *arXiv preprint arXiv:1607.01794*, 2016.